

CLAIMS

What is claimed is:

1. A system for enabling socket communication in COBOL program, comprising:
 - a memory block;
 - a COBOL program communicating with the memory block;
 - a socket; and
 - a COBOL routine callable from the COBOL program, the COBOL routine operable to read information from the socket and write the information read from the socket to the memory block in response to the COBOL program call.
2. The system of Claim 1, wherein the COBOL program further communicates with the COBOL routine to initiate the COBOL routine communication with the socket and the memory block.
3. The system of Claim 1, wherein the COBOL routine is further defined as a subroutine of the COBOL program.
4. The system of Claim 1, wherein the COBOL routine is further defined as a COBOL library having a plurality of routines callable by the COBOL program.
5. The system of Claim 1, wherein the COBOL routine is further defined as a compiler enabled function usable by the COBOL program.

6. Method for enabling socket communication in COBOL program, comprising:
 - requesting, by a COBOL program, information from a socket;
 - retrieving, by a COBOL routine, information from the socket;
 - writing, by the COBOL routine, information read from the socket to a memory block; and
 - reading from the memory block, by the COBOL program, the information.
7. The method of Claim 6, wherein the method further comprises managing, by the COBOL routine, a connection with the socket.
8. The method of Claim 7, wherein managing includes listening on the socket connection.
9. The method of Claim 7, wherein managing includes disconnecting the connection with the socket.
10. The method of Claim 6, wherein the method further comprises, establishing, by the COBOL routine, a connection with the socket.
11. The method of Claim 10, wherein the connection with the socket is established in response to a request from the COBOL program
12. The method of Claim 6, wherein the COBOL routine provides an address to the COBOL program, the address identifying a location of the memory block where the information is written.

13. The method of Claim 12, wherein the method further comprises mapping, by the COBOL program, the memory block into the COBOL program.
14. The method of Claim 13, wherein the mapping is accomplished using a copybook.
15. The method of Claim 6, wherein the information is provided in an EBCDIC format and wherein the method further comprises converting the information from the EBCDIC format to an ASCII format.
16. The method of Claim 15, wherein the conversion is accomplished by the COBOL routine.
17. The method of Claim 6, wherein the COBOL routine further includes a coordination module to coordinate such that the COBOL routine only reads when the socket has information and only writes when the socket is not full.
18. The method of Claim 6, further comprising initiating a call to the operating system by the COBOL routine to establish a socket connection.
19. The method of Claim 18, wherein the call to the operating system is further defined as a bit-level call to the operation system of a mainframe computer system.
20. The method of Claim 19, wherein the COBOL routine is further defined as written in COBOL programming language.

21. A system for enabling pipe communication in COBOL program, comprising:
- a memory block;
 - a COBOL program communicating with the memory block;
 - a pipe; and
 - a COBOL routine callable from the COBOL program, the COBOL routine operable to read information from the pipe and write the information read from the pipe to the memory block in response to the COBOL program call.
22. The system of Claim 21, wherein the memory block is further defined as a mainframe memory block and wherein the COBOL program and COBOL routine are operable on a mainframe computer system.
23. The system of Claim 22, wherein the COBOL routine is further defined as a COBOL technical layer having a plurality of routines callable by the COBOL program, the COBOL technical layer including:
- a create module communicating with a computer system and operable to create a pipe connection;
 - a connect module operable to promote attachment to the pipe connection;
 - an open module operable to open the pipe connection to promote communication via the pipe connection;
 - a write module operable to write information to the pipe connection, the write module operable to verify that the pipe connection is not full prior to writing information and further operable to block when the pipe connection is full;

a read module coupleable to the pipe connection to read information from the pipe connection;

a release module to release the pipe connection;

a remove module to remove the pipe connection from the computer system; and

a delete module to delete the pipe connection wherein the pipe connection is closed.

24. A method for socket communication in COBOL, comprising:
- reading, by a routine, information from a socket;
 - writing, by the routine, the information to an area; and
 - reading, by a COBOL program, the information from the area, the COBOL program and the routine operating in the same runtime environment.
25. The method of Claim 24, wherein the area is a file.
26. The method of Claim 24, wherein the area is a memory area.